CSC 4410: Operating Systems
(Spring 2023)

Project 1: Blocking Queue

Assigned: Fri Feb 17 2023
Due: 10:00:00 AM on Mon Mar 20 2023
Team Size: 1-3
Language: Java
Out of: 175 points

In this project, you will implement a blocking queue, which can handle multi-threaded calls to the add and remove methods. Important rules about what you're allowed to use/reference:

- You are not allowed to use any already-defined concurrent data structures, neither built-in to standard Java packages nor available online. You are solving the concurrency here. If you violate this, you will get a zero for the project.
- You are not allowed to look up how to implement a concurrent queue or blocking queue or thread-safe queue in any programming language (including pseudocode). If you're stuck on something, review the lecture notes or come ask me questions. Violating this counts as a breach of the honor code and may incur administrative and academic penalties.
- You are expected to use your MySemaphore.java code from the prior project.

Part 0, 0 points: Create a new Java file, `MyBlockingQueue.java`. Include a file header that includes the names of all teammembers. I recommend doing this as a Javadoc comment:

```
/**
 * Models a thread-safe BlockingQueue.
 *
 * @author Barbara M. Roberts <barbie@mattel.com>
 */
```

Part 1, 0 points: Start your MyBlockingQueue class and, in good Java practice, include a generic type for the type of objects that will be stored in your queue. In case you haven't done that before, you can do it like this:

```
public class MyBlockingQueue<YourGenericType> {
```

But you should choose your own GenericType. It acts like a variable for a type.

Part 2, 0 points: Add a constructor that takes a single int parameter. This will be the maximum number of elements that can be inside your MyBlockingQueue at any time. Think about what fields you want your class to have and do all the appropriate initialization in the constructor. (You'll likely have to come back here and change things as you continue through the project.) You can then create blocking queues like this:

```
MyBlockingQueue<String> blockingStrings = new MyBlockingQueue<>(5);
```

Part 3, 0 points: Add a method, `add`, that takes a parameter of your generic type and adds it to the back of your queue. The signature should start off looking like this:

```
public void add(YourGenericType element) {
```

Part 4, 10 points: Add a `toString` method that prints out the elements in the queue.

Part 5, 50 points: Add a method, `remove`, that returns the element at the front of the queue.

Part 6, 7 points: Add a method, `getNumElements`, that returns the number of elements currently stored in the queue.

Part 7, 8 points: Add a method, `getFreeSpace`, that returns the number of empty spots in the queue.

Part 8, 0 points: At this point, you should be able to compile and run your queue with my code: BlockingQueueTester.java. Most of the project points are indicated there, though there isn't a perfect test for every part!

Part 9, 10 points: Make sure that your queue is FIFO (First-In, First-Out).

Part 10, 30 points: Make sure that your remove method blocks appropriately: when there are no elements in the queue.

Part 11, 30 points: Make sure that your add method blocks appropriately: when the queue is full.

Part 12, 10 points: Make sure that your queue passes my stress tests. (There are some commented-out print statements you can comment-in as necessary while you're testing.)

Part 13, 20 points: Make sure that there are no race conditions or possible deadlocks in your project. You may or may not see this during testing, so be very careful. I will look carefully at your code!

Submitting your Project:

Choose a team name that no one else will choose. Put all the source code files into a folder named *teamName*Project1 then zip the folder into a .zip or a .tar.gz file (*no other file formats will be accepted*). Finally, upload the zipped folder to the Canvas assignment. If you fail to follow these directions, I will ask you to redo it and you will lose points as though you hadn't turned it in until I get the resubmission. I'll be looking for the following files in your folder:

- `MySemaphore.java`
- `MyBlockingQueue.java`